**What are lists and tuples? What is the key difference between the two?**

**Lists**and **Tuples** are both s**equence data types** that can store a collection of objects in Python. The objects stored in both sequences can have **different data types**. Lists are represented with **square brackets** ['sara', 6, 0.19], while tuples are represented with **parantheses** ('ansh', 5, 0.97).  
But what is the real difference between the two? The key difference between the two is that while **lists are mutable**, **tuples** on the other hand are **immutable** objects. This means that lists can be modified, appended or sliced on the go but tuples remain constant and cannot be modified in any manner. You can run the following example on Python IDLE to confirm the difference:

### What is pass in Python?

The pass keyword represents a null operation in Python. It is generally used for the purpose of filling up empty blocks of code which may execute during runtime but has yet to be written. Without the **pass** statement in the following code, we may run into some errors during code execution.

### What are modules and packages in Python?

Python packages and Python modules are two mechanisms that allow for **modular programming** in Python. Modularizing has several advantages

**Modules**, in general, are simply Python files with a .py extension and can have a set of functions, classes, or variables defined and implemented. They can be imported and initialized once using the import

**Packages** allow for hierarchial structuring of the module namespace using **dot notation**. As, **modules** help avoid clashes between global variable names, in a similar manner, **packages** help avoid clashes between module names.

**How can we define private member in a class?**

**Private**attributes are attributes with double underscore prefixed to their identifier eg. \_\_ansh. They cannot be accessed or modified from the outside directly and will result in an AttributeError if such an attempt is made

### What is docstring in Python?

Documentation string or docstring is a multiline string used to document a specific code segment. The docstring should describe what the function or method does

### What is slicing in Python?

As the name suggests, ‘slicing’ is taking parts of.

Syntax for slicing is **[start : stop : step]**

### What is Scope Resolution in Python?

Sometimes objects within the same scope have the same name but function differently. In such cases, scope resolution comes into play in Python automatically. A few examples of such behavior are: Python modules namely 'math' and 'cmath' have a lot of functions that are common to both of them - log10(), acos(), exp() etc. To resolve this ambiguity, it is necessary to prefix them with their respective module, like math.exp() and cmath.exp()

### What are decorators in Python?

**Decorators** in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented the @decorator\_name in Python and are called in a bottom-up fashion.

### What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic sugar** constructs that help **build altered** and **filtered lists**, dictionaries, or sets from a given list, dictionary, or set. Using comprehensions saves a lot of time and code that might be considerably more verbose (containing more lines of code). Let's check out some examples, where comprehensions can be truly beneficial:

my\_list = [2, 3, 5, 7, 11]

squared\_list = [x\*\*2 **for** x **in** my\_list] # list comprehension

# output => [4 , 9 , 25 , 49 , 121]

squared\_dict = {x:x\*\*2 **for** x **in** my\_list} # dict comprehension

# output => {11: 121, 2: 4 , 3: 9 , 5: 25 , 7: 49}

### What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period. Lambda functions can be used in either of the two ways:

mul = **lambda** a, b : a \* b

print(mul(2, 5)) # output => 10

### How Python is interpreted?

* Python as a language is not interpreted or compiled. Interpreted or compiled is the property of the implementation. Python is a bytecode(set of interpreter readable instructions) interpreted generally.
* Source code is a file with .py extension.
* Python compiles the source code to a set of instructions for a virtual machine. The Python interpreter is an implementation of that virtual machine. This intermediate format is called “bytecode”.
* .py source code is first compiled to give .pyc which is bytecode. This bytecode can be then interpreted by the official CPython or JIT(Just in Time compiler) compiled by PyPy.

### What does \*args and \*\*kwargs mean?

**\*args**

* \*args is a special syntax used in the function definition to pass variable-length arguments.
* “\*” means variable length and “args” is the name used by convention. You can use any other.

def multiply(a, b, \*argv):

mul = a \* b

for num in argv:

mul \*= num

return mul

print(multiply(1, 2, 3, 4, 5)) #output: 120

**\*\*kwargs**

* \*\*kwargs is a special syntax used in the function definition to pass variable-length keyworded arguments.
* Here, also, “kwargs” is used just by convention. You can use any other name.
* Keyworded argument means a variable that has a name when passed to a function.
* It is actually a dictionary of the variable names and its value.

def tellArguments(\*\*kwargs):

for key, value in kwargs.items():

print(key + ": " + value)

tellArguments(arg1 = "argument 1", arg2 = "argument 2", arg3 = "argument 3")

### 

### What are negative indexes and why are they used?

* Negative indexes are the indexes from the end of the list or tuple or string.
* These indexes used in reverse indexing are called negative indexes.

### What is init method in python?

The **init** method works similarly to the constructors in Java. The method is run as soon as an object is instantiated. It is useful for initializing any attributes or default behaviour of the object at the time of instantiation.

### What are some of the most commonly used built-in modules in Python?

Os, math, sys, random, re, datetime, JSON

### List down 5/3 method of :

### List, Dictionary, string?

### Define PIP?

### PIP stands for Python Installer Package. As the name indicates, it is used for installing different python modules. It is a command-line tool providing a seamless interface for installing different python modules. It searches over the internet for the package and installs them into the working directory without the need for any interaction with the user.

pip install <package\_name>

### What is main function in python? How do you invoke it?

### ****What is pep 8?****

PEP stands for **Python Enhancement Proposal.**It is a set of rules that specify how to format Python code for maximum readability.

### ****Is python case sensitive?****

Yes. Python is a case sensitive language.

### ****What is the usage of help() and dir() function in Python?****

1. **Help() function**: The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, etc.
2. **Dir() function**: The dir() function is used to display the defined symbols.

### ****Is python numpy better than lists?****

We use python numpy array instead of a list because of the below three reasons:

1. Less Memory
2. Fast
3. Convenient

### ****When will the else part of try-except-else be executed?****

when no exception occurs

### ****What is a dictionary in Python?****

Python dictionary is one of the supported [data types in Python](https://intellipaat.com/blog/tutorial/python-tutorial/python-datatypes/). It is an unordered collection of elements. The elements in dictionaries are stored as key-value pairs. Dictionaries are indexed by keys.

**Difference between Filter and Map function.**

### ****What is the difference between append() and extend() methods****

Both append() and extend() methods are methods used to add elements at the end of a list.

**append**(element): Adds the given element at the end of the list that called this append() method

**extend**(another-list): Adds the elements of another list at the end of the list that called this extend() method

**Why do we need a break in Python?**

**Answer: Break helps in controlling the Python loop by breaking the current loop from execution and transfer the control to the next block.**

**Why do we need a continue in Python?**

**Answer: A continue also helps in controlling the Python loop but by making jumps to the next iteration of the loop without exhausting it.**

**Can we use a break and continue together in Python? How?**

**Answer:  Break and continue can be used together in Python. The break will stop the current loop from execution, while jump will take to another loop.**

**Does Python support an intrinsic do-while loop?**

**Answer: No Python does not support an intrinsic do-while loop.**

**Why do we need membership operators in Python?**

**Answer:  We need membership operators in Python with the purpose to confirm if the value is a member in another or not.**

#### **Define self in Python**

**Answer:**An instance of a class or an object is self in Python. It is included as the first parameter. It helps to differentiate between the methods and attributes of a class with local variables.